Published in final edited form as:

J Chem Theory Comput. 2017 September 12; 13(9): 4593–4603. doi:10.1021/acs.jctc.7b00690.

Coarse-grained Directed Simulation

Glen M. Hocky, **Thomas Dannenhoffer-Lafage**, and **Gregory A. Voth**[*]

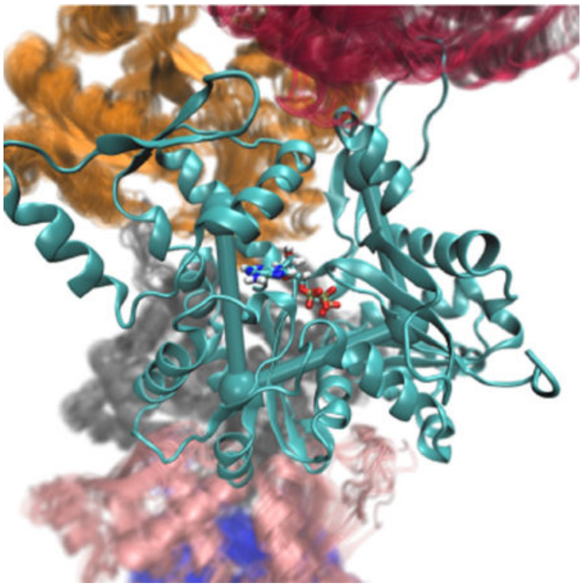
Department of Chemistry, James Franck Institute, and Institute for Biophysical Dynamics, The University of Chicago, 5735 South Ellis Avenue, Chicago, Illinois 60637, United States

Abstract

Many free-energy sampling and quantum mechanics/molecular mechanics (QM/MM) computations on protein complexes have been performed where, by necessity, a single component is studied isolated in solution while its overall configuration is kept in the complex-like state by either rigid restraints or harmonic constraints. A drawback in these studies is that the system's native fluctuations are lost, both due to the change of environment and the imposition of the extra potential. Yet, we know that having both accurate structure and fluctuations is likely crucial to achieving correct simulation estimates for the subsystem within its native larger protein complex context. In this work, we provide a new approach to this problem by drawing on ideas developed to incorporate experimental information into a molecular simulation by relative entropy minimization to a target system. We show that by using linear biases on coarse-grained (CG) observables (such as distances or angles between large subdomains within a protein), we can maintain the protein in a particular target conformation while also preserving the correct equilibrium fluctuations of the subsystem within its larger biomolecular complex. As an application, we demonstrate this algorithm by training a bias that causes an actin monomer (and trimer) in solution to sample the same average structure and fluctuations as if it were embedded within a much larger actin filament. Additionally, we have developed a number of algorithmic improvements that accelerate convergence of the on-the-fly relative entropy minimization algorithms for this type of application. Finally, we have contributed these methods to the PLUMED open source free energy sampling software library.

TOC image

[*]Corresponding Author: gavoth@uchicago.edu.



Introduction

It is a fundamental challenge of molecular dynamics (MD) simulation that the accessible time and length scales are limited by the level of detail at which a system is described.[1] As a consequence, computational studies generally compromise on the size of the system under consideration in order to achieve a desired level of detail. A prime example arises in the study of enzymes where reactivity must be incorporated into the simulation, but it is thus far impractical to treat the entire system quantum mechanically. Hence, in order to do a quantum calculation on the active site while including the important role of the protein environment, hybrid methods have been developed to couple the protein's fluctuations into the reactive subsystem. These ideas led to the commonly used practice of QM/MM simulation, and contributed to the 2013 Nobel Prize in Chemistry.[2]

The continued development of enhanced sampling techniques, as well as advances in computational power, have made it possible to compute complex free energy surfaces for quantum mechanical reactions or conformational transitions taking place in biomolecular systems (see, e.g., Refs. 3–4). However, many problems of current focus do not occur isolated in solution, but rather in a much larger macromolecular context. For example, we might be interested in a process occurring for a particular protein embedded in a membrane or one within a much larger many-protein complex. In these cases, free energy sampling methods, which aim to enhance sampling along "slow" degrees of freedom in a low dimensional set of collective variables (CVs), are constrained by our ability to sample the many other degrees of freedom in the system. In the past, such simulations have been made to be more computationally tractable by extracting the key subcomponent of the system and adding restraints or harmonic biases to keep this subsystem in approximately the same configuration as it would sample in its native (much larger) context. Unfortunately, doing so neglects both the effect of long range forces coming from the other molecules in the complex, and alters the fluctuations sampled by the subsystem, which in many cases are strongly coupled with dynamics in the larger system. In this work, we focus on the latter problem, namely, we wish to have a protein of interest isolated in solution sample configurations as if it were embedded within a larger multi-protein complex, thus greatly reducing the computational cost and hence enabling studies where extensive free energy sampling is possible. In many cases, the larger complex can also first be simulated for some limited amount of MD sampling time in order to obtain information about the conformational ensemble of the subsystem of interest (alternatively, one could imagine approximating this information from normal mode or elastic network model analysis performed on experimental structures[5]).

In an ideal world with perfect sampling, one could simulate the subsystem by first finding an exact potential of mean force, by integrating over all the other configurations of the larger system. In the canonical ensemble, for a subsystem with N particles having the coordinates $q = (q_1, q_2, \ldots, q_{3N})$ and a supersystem with $N+M$ particles with coordinates $r = (q_1, q_2, \ldots, q_{3N}, q_{3N+1}, \ldots, q_{3N+3M})$ and overall potential energy function $U(r)$, this corresponds to first calculating:

$$F(X) = -k_B T \ln\left(\frac{\int dr \delta(q-X) e^{-\frac{U(r)}{k_B T}}}{\int dr\, e^{-\frac{U(r)}{k_B T}}}\right) \quad (1)$$

This many-body potential of mean force (PMF), $F(X)$, captures the effects of the exact coupling to the larger supersystem on the subsystem variables $X = q$, and could be used to measure the average of any desired observable function of the subsystem variables, $\langle f \rangle$, as if we had performed the computation on the supersystem:

$$\langle f \rangle = \frac{\int dr\, f(q) e^{-\frac{U(r)}{k_B T}}}{\int dr\, e^{-\frac{U(r)}{k_B T}}} = \frac{\int dX\, f(X) e^{-\frac{F(X)}{k_B T}}}{\int dX\, e^{-\frac{F(X)}{k_B T}}} \quad (2)$$

However, we know in practice this cannot be done for anything but the simplest cases, both because the amount of sampling required would be enormous and because the amount of information required to express the many-body PMF $F(X)$ is too large.

In this work, we suggest an alternative approach and show that some key information about the supersystem can be imparted to the subsystem in a different and computationally practical way. Here, we will modify the potential energy function that would normally be used for the subsystem alone, by “learning” a bias function on coarse-grained (CG) observables of interest, f_i, via a relative entropy based approach. The bias is such that when those observables are sampled by the biased subsystem they will match what would be observed by sampling from $F(X)$.

The idea of adding a bias to incorporate additional information into a system’s Hamiltonian comes from work where experimental data is included into an MD simulation via either adding a set of linear biases to the system’s Hamiltonian or, alternatively, using many copies of the system biased to have an ensemble averaged target observable.[6–7] These techniques have been shown to minimally bias the system such that it samples the target values of the observables.[6] Several techniques now exist to determine a bias to match experimental data on-the-fly,[8–9] starting with a method called Experiment Directed Simulation (EDS), where the linear restraints are learned through a stochastic gradient descent procedure.[8] Later work[10] demonstrated that this type of EDS bias always decreases (or at least, does not increase) the relative entropy with respect to an ideal target distribution that gives the same desired properties, and so this method can be used to systematically improve multiscale CG (MS-CG) models that may utilize an imperfect molecular force field. This method of adaptive linear biases is also beginning to demonstrate its utility in other applications, including providing a significant improvement of the static and dynamic properties of an *ab initio* MD (AIMD) water system which is based on a rather inaccurate level of electronic density functional theory, biasing the oxygen-oxygen radial distribution function.[11] A similar method was used to incorporate experimental data (and additionally, experimental errors) to improve the quality of force fields for RNA.[9] The empirical evidence in all cases is that the biasing of one observable tends to improve (or at least not decrease) the quality of others which were not biased, as would be expected for this class of minimal bias techniques.

In this work, we test whether the methods developed in EDS can learn biasing parameters that constrain a subsystem to behave as if it is embedded in its native (much larger) supersystem environment, recapitulating some of the desired properties from Eqs. 1 and 2 above. We apply these ideas to the protein actin, because it is a challenging biopolymer target with complex structural transitions that have nevertheless been relatively well characterized in MD simulation and experiment by our group and others. In solution, actin exists as a globular domain (G-actin) with a bound ATP molecule (Figure 1A)[12–13] and adopts a twisted conformation characterized by a dihedral angle of its four sub-domains $\sim -20°$.[14] Monomers can assemble to form a non-covalent semi-flexible biopolymer (F-actin), within which each subunit is flattened, adopting a dihedral angle $> -10°$. This flattening is associated with an increase in rate of actin catalyzed hydrolysis of the bound ATP molecule by a factor of $> 10^4$,[3, 15] such that the hydrolysis rate is on the order of ~1 sec^{-1}. The release of the free inorganic phosphate is very slow, occurring on a time scale of ~5 min,[16] and makes the actin filament softer and more prone to depolymerization.[17–20] The hydrolysis and phosphate release are crucial processes governing the lifetime and structural properties of actin filaments and cytoskeletal networks in cells, and the molecular mechanisms have been studied using simulations. While it is now relatively standard to simulate with MD a semi-periodic actin filament consisting of 13 subunits (~500,000 atoms when solvated with water), the extensive simulation time needed to study phosphate release by free energy methods and the QM/MM methods required to study the explicit hydrolysis reaction preclude simulating such a large system; instead, only a single actin monomer has been used while being restrained in the filamentous or globular form.[3–4]

Below, we will show that using the EDS approach on a set of CG observables (a combination we call Coarse-grained Directed Simulation, or CGDS) can be used to minimally bias an actin monomer to be in a filamentous-like configuration while maintaining correct fluctuations for the two collective variables that characterize the transition from globular to filamentous actin structure.[13, 21–22]. However, given the large and complex system size compared to what has been studied previously, we found that the previous algorithms did not learn EDS biasing parameters fast enough to achieve these goals. Hence a major part of the present work is devoted to improving the algorithms for this type of problem so they can be practicable for similar future applications. These enhancements are first developed on a CG model of actin for speed of testing and development, then demonstrated on fully atomistic systems. All of these algorithms have been implemented and are available for use in the open source sampling library PLUMED2,[23] and the major components are already included in the main release of the PLUMED2 software as an optional module.

The remaining sections of this paper is organized as follows:

1. Methods: The construction both of atomistic systems and CG systems are described. The CGDS relative entropy minimization algorithm is written in a general framework that encompasses both prior work and our changes to the algorithms.

2. Results: A CG model is used to show how the prior methods can be optimized, and the performance of a first order guess for bias parameters. We then show how the optimization algorithm can be further improved for multiple CVs by transitioning to a simultaneous update of the bias parameters rather than a stochastic one, in this case. These improved algorithms are then demonstrated to work for a monomeric as well as trimeric actin system which would be appropriate for future free energy simulations.

3. Discussion and Conclusions: Future outlook and ramifications, as well as the challenges encountered are discussed.

Methods

Molecular dynamics simulations of actin filaments and monomers

G-actin with a bound ATP and a periodic 13 subunit F-actin structure with bound ATP were built and equilibrated at 310 K as described previously[21–22, 24] (~94,000 atoms and ~485,000 atoms respectively), with the structure of ATP-bound actin derived from the crystal structure PDB ID 1NWK[25] and for F-actin from the electron microscopy structure in PDB ID 2ZWH.[14] For the filament, the actin subunit had its nucleotide replaced by an ATP, bound magnesium, and waters within 5 angstroms of the nucleotide from a previously equilibrated monomer simulation. MD simulation of these structures was then performed using GROMACS[26] for ~5 ns to relax the configurations before the data shown below. A third and fourth system were then created by solvating a single actin monomer (~94,000 atoms) and a trimer of actin monomers (~138,000 atoms) from the equilibrated filament structure and relaxing those structures for ~5 ns.

Construction of an elastic network model for an actin monomer

An MD trajectory of the G-actin monomer bound to ATP was used to generate a CG elastic network model of an actin monomer as follows: after equilibrating the structure for 20 ns, the next 50 ns of simulation data were mapped onto a previously characterized 12-bead representation of an actin monomer, with beads 1–4 representing the four major subdomains of actin, seven others comprised by other important sub-regions, and a final bead for the nucleotide (Figure 2A);[21–22] a heterogeneous elastic network model (hENM) was then built from this trajectory using the method of Ref. 27. In brief, all pairs of beads closer than 100Å were connected by an effective spring whose rest length was given by the average separation in the MD trajectory and whose spring constants are all identical at first. An iterative procedure was then performed that updates the values of the spring constants by an amount proportional to the difference between the normal mode fluctuations along that bond in a given iteration and the fluctuations of that pair distance in the mapped MD simulation (summed over $3N_{\text{beads}}$ normal modes). MD simulations of this CG model then reproduce approximately the structural ensemble observed in the original all-atom trajectory.[27] This hENM model was then used to quickly test improvements in the methods described next.

Relative entropy minimization

In this work, the objective is for our system to evolve under minimally biased dynamics such that average value of particular CG observables of a subsystem of interest match target values obtained via simulations of a larger encompassing supersystem (or alternatively values obtained from experiment). In principle, as in previous work, we can derive the necessary change to the system Hamiltonian (H_0) by minimizing the relative entropy between the distribution normally sampled by that system, $P_0(X)$, and the distribution, $P(X)$, arising from an unknown Hamiltonian (H). The latter unknown Hamiltonian system is subject to constraints on a mean of a set of observables, $\{f_i(X)\}$, which are scalar functions of the configuration of the system and are known properties of the system described by the unknown Hamiltonian.[8, 10] In other words, we want to minimize the functional

$$S[P(X)]=\int dXP(X)\ln\left(\frac{P(X)}{P_0(X)}\right) \quad (3)$$

with the constraints $\int dX\, P(X) = 1$, and $\int dX\ f_i(X)P(X)=\overline{f}_i$. This is formally solved by introducing Lagrange multipliers, $\{\lambda_i\}$:

$$P(X)=P_0(X)e^{\beta(\lambda_0+\sum_i\lambda_i f_i(X))} \quad (4)$$

with $\beta=1/k_B{}^T$. Using the first normalization constraint to set λ_0, and taking P_0 from the canonical ensemble gives the result:

$$P(X)=\frac{e^{-\beta(H_0(X)+\sum_i\lambda_i f_i(X))}}{\int dXe^{-\beta(H_0(X)+\sum_i\lambda_i f_i(X))}} \quad (5)$$

By comparison with the usual canonical ensemble distribution function, we see here that in order to simulate our system and have it maintain particular target values of our observables, $\left\{\overline{f}_i\right\}$, i.e., that are manifest in a simulation (or experiment) for the full and larger encompassing supersystem, we must modify the Hamiltonian H_0 of our system to include a linear term for each observable with (at this point) unknown proportionality constants $\{\lambda_i\}$ (these are the same as the $\{\alpha_i\}$ in Ref. 10 and $\{\lambda_i\}$ in Ref. 9, but differ by a constant factor $\left\{\overline{f}_i\right\}$ from $\{\alpha_i\}$ in the EDS formulation in Refs. 8 and 11).

Several papers have offered suggestions for how to determine these parameters in a molecular simulation context.[6–9, 28] In each case, the idea is to iteratively update the values of $\{\lambda_i\}$ proportional to the difference $\Delta_i(\boldsymbol{\lambda}^t)=\langle f_i(X)\rangle_t-\overline{f}_i$, where the average is taken by sampling for a time τ^{avg} using the Lagrange multipliers from the t^{th} iteration. The rule for updating $\boldsymbol{\lambda} = \{\lambda_i\}$ (bold font indicating vectors, double-bar indicating matrices, dots indicating vector matrix multiplication, other vector arithmetic is element-wise) then takes a form similar to:

$$\boldsymbol{\lambda}^{t+1}=\boldsymbol{\lambda}^t-\boldsymbol{\eta}^t\left(\boldsymbol{\Delta}(\lambda^t)\cdot\left(\frac{\partial\boldsymbol{\Delta}}{\partial\boldsymbol{\lambda}^t}\right)^n\right)\equiv\boldsymbol{\lambda}^t-\boldsymbol{\eta}^t\boldsymbol{\delta}^t, \qquad (6)$$

where $n = 1$ corresponds to gradient descent,[8] $n = 0$ a scheme that only depends on the current distance from the average observables,[9] and $n = -1$ to Newton's method, and the error function being function being minimized is the total squared difference from observed to target CV values, i.e. $\varepsilon_t = \Sigma_i \Delta_i(\boldsymbol{\lambda}^t)^2$.[7] The way that the pre-factor $\boldsymbol{\eta}^t$ is adjusted corresponds to a learning rate rule, that is, how much trust to ascribe to the step size coming from the other λ-dependent terms. For example, Bussi and coworkers use[9]:

$$\eta_i^t=\frac{A_i}{1+\frac{t}{\tau_i}} \qquad (7)$$

while White and Voth use[8, 10–11]:

$$\eta_i^t=\frac{A_i}{\sqrt{\sum_{j=1}^{t}\left(\delta_i^t\right)^2}} \qquad (8)$$

For the linear bias, the gradient term is a matrix with entries given by:

$$\left(\frac{\partial\boldsymbol{\Delta}}{\partial\boldsymbol{\lambda}^t}\right)_{ij}=-\langle f_i f_j\rangle_t+\langle f_i\rangle_t\langle f_j\rangle_t, \qquad (9)$$

so, the gradient is proportional to the covariance of the two observables on iteration t:

$$\left(\frac{\partial \boldsymbol{\Delta}}{\partial \boldsymbol{\lambda}^t}\right) = -\,\mathrm{Cov}^t(f_i, f_j) \equiv \overline{\overline{J}} \qquad (10)$$

In all cases where these methods have been applied in a molecular context, a stochastic procedure was used where one Lagrange multiplier was chosen randomly and adjusted based on this recipe (using only the diagonal elements in Eq. 9). Below, we will use the learning rate rule of White and Voth, but show that for the type of problems we are interested in, rather than the stochastic gradient descent (SGD) a simultaneous adjustment of all the parameters using the full covariance, either with gradient descent (Covar) or a Levenberg-Marquardt-type algorithm (LM) as suggested previously (see below)[28] can be superior, and prevent the need for tuning of the constant factor A_i for each observable. (Note that in the work of White and Voth and in this implementation, the constant A_i is scaled by the target observable value, so the A_i in this general formulation is replaced by $A_i = 2A_i^{'}/\overline{f}_i$, with $A_i^{'}$ the value set as a RANGE parameter in the current Plumed2 implementation. The bias parameters λ_i always have units [Energy]/[CV], so in this way, $A_i^{'}$ has units of [Energy]. These are the values reported simply as A below.)

In the LM algorithm, the step size in Eq. 6 is replaced by

$$\boldsymbol{\delta}^t = \left(\left[\overline{\overline{J}}^T \overline{\overline{J}} + \gamma^t \mathrm{diag}\left(\overline{\overline{J}}^T \overline{\overline{J}}\right)\right]^{-1} \overline{\overline{J}}^T\right) \cdot \boldsymbol{\Delta}(\boldsymbol{\lambda}^t) \qquad (11)$$

where $\mathrm{diag}\left(\overline{\overline{J}}^T \overline{\overline{J}}\right)$ is the purely diagonal matrix having the same diagonal elements as $\overline{\overline{J}}^T \overline{\overline{J}}$, and γ^t is a mixing parameter (which in general could be made adaptive as in Ref. 28), and causes the method to behave with some character of Newton's method ($\gamma = 0$) and steepest-descent ($\gamma \gg 1$).[28] We have implemented an adaptive version of the algorithm where we average the total squared error ε_t for all CVs over *LM_stride* iterations of the algorithm. This quantity is retained over m windows of length $LM\_stride * \tau^{\mathrm{avg}}$. If the error decreases monotonically for these m stages, γ is increased by a multiplicative factor $l > 1$, i.e. $\gamma^{t+1} = l\gamma^t$. If it decreases in each stage, $\gamma^{t+1} = \gamma^t / l$. If it is not monotonic over m stages, then γ is not changed.

All algorithmic parameters for each figure are listed in Table 1. We make a practical note here, that in order to reduce the overhead of biasing via PLUMED, CVs are defined using only two atoms per protein residue (C_α, C_β) and for all-atom simulations we use the built-in multiple time stepping procedure of Bussi and coworkers,[29] with the bias algorithm performed every-other MD time step (STRIDE=2).

Results

Actin monomers in solution are more twisted and open, with larger fluctuations than subunits in a filament

As described in the methods section, MD simulations of an actin filament and actin monomers from two different starting configurations were performed. The difference between the G-actin and F-actin structure can be well characterized (as in previous works[13, 21]) by defining the cleft distance (d) and twist angle (ϕ) as shown in Figure 1A. In this study, CG bead positions used for analysis and for biasing are defined by computing the center of mass of the C_α and C_β in each actin subdomain. Using these definitions, we computed the value of actin monomer twist angle and cleft distances in our MD simulations. In Figure 1B, we show that, as in previous simulations and experimental studies, an actin subunit in a filament is flat ($\phi \geq -10°$) and closed ($d \lessapprox 21$Å)[13–14] while a G-actin with bound ATP is twisted, ($\phi \lessapprox -20°$) and more open ($d \gtrapprox 25$Å).[22, 25] Interestingly, in our simulation of an ATP-bound monomer starting from a filamentous configuration, the monomer readily adopts a G-actin like structure over the course of a relatively short MD simulation, suggesting that the twisted/open structure has a lower free-energy outside of a filamentous context, as expected based on known actin biology.[13] Finally, we observe that the fluctuations of these two CVs for an actin monomer in solution are larger than in a filament, where allosteric coupling to adjacent actin subunits constrains the range of conformations (see Table 2).

Linear bias method can out-perform harmonic bias for matching target CV value

Unconstrained simulations of our hENM test system show a ϕ angle oscillating around the atomistic G-actin value from Figure 2B. Applying a harmonic bias (HB) to the hENM model of the form $\frac{1}{2}k_\phi\left(\phi - \overline{\phi}\right)^2$ with spring constant $k_\phi = 10^3$ kJ/mol/rad^2 and even $k_\phi = 10^4$ kJ/mol/rad^2 centered at $\overline{\phi} = -6.3°$ moves the sampled value of ϕ closer to but not all the way to this target value. As discussed previously, applying this bias also substantially changes the size of fluctuations of this observable (see Table 2).[6] In contrast, Figure 2C shows that when using the adaptive algorithm of Ref. 8, the target average is achieved exactly, although it takes some time to learn the bias parameter.

Adaptive algorithm has optimal averaging time for 1 CV

The adaptive algorithms discussed in the Methods section sample observable values over a time τ^{avg} before adjusting the bias parameter based on a learning rule. In Figure 1C we show in simulations of the hENM actin model that the convergence of the gradient descent algorithm can be very sensitive to the value of τ^{avg}, and appears to have an optimal value, in this case approximately 10–15 ps. Very short averaging windows produce a poor average of the variance of the observable used in the update rule, and very long averaging time produces a better average, but too long is spent on this process. An idea of how to set this can come from computing the autocorrelation function (ACF) for the CV in an unbiased simulation. For the twist angle, we estimate the autocorrelation time (when the ACF decays to $1/e$) at ~ 4 ps. Hence to get a good estimate for the average and variance of this CV used in the algorithm, it seems that at least 2 autocorrelation times are needed. The speed of convergence can also be adjusted by changing the constant A in the learning rule (in this case all simulations used $A = 10\ k_BT$), but we have found that having too large of an initial value of A can cause the bias parameters to overshoot, hence it is better to choose good values for τ^{avg} first, as this has a non-linear effect on the learning time.

Derivation and validity of a linear response approximation to bias parameters

If the target value for a CV is close to the value in an unbiased simulation, then we know that λ_i for that CV will be small. Hence, we can expand the exponential as a Taylor series around $\lambda_i = 0$. We will illustrate this derivation first for a single CV. From Eq. 5, we can see that when the correct Lagrange multiplier has converged for a CV f':

$$\overline{f'}=\left\langle f'\right\rangle_{\lambda}=\frac{\int dX f'(X)e^{-\beta(H_0(X)+\lambda f'(X))}}{\int dX e^{-\beta(H_0(X)+\lambda f'(X))}} \qquad (12)$$

Without loss of generality, we can consider the CV $f \equiv f' - \overline{f'}$, with target value $\overline{f}=0$. Hence for this new CV, the left-hand side of this equation is zero and the denominator on the right-hand side, as a constant, does not affect the equality. We can then expand the exponential in the numerator and write:

$$0=\int dXf(X)e^{-\beta H_0(X)}\left(1-\beta\lambda f(X)+\frac{(-\beta\lambda)^2}{2}f^2(X)+\cdots\right) \qquad (13)$$

Dividing both sides of the equation by the unbiased partition function $Z=\int dXe^{-\beta H_0(X)}$, we can express each term as an average that can be obtained in the unbiased simulation. This gives an equation to be solved for the unknown constant λ.

$$0=\langle f\rangle-\beta\lambda\left\langle f^2\right\rangle+\frac{(-\beta\lambda)^2}{2}\left\langle f^3\right\rangle+\cdots=\sum_{j=0}^{\infty}\frac{(-\beta\lambda)^j\left\langle f^{j+1}\right\rangle}{j!} \qquad (14)$$

If the averages $\langle f^j\rangle$ can be computed accurately from an unbiased simulation, then this equation can be truncated at some power and solved for λ, using, for example, Mathematica.[30] At first order, this equation is trivially solved:

$$\lambda=k_BT\frac{\langle f\rangle}{\langle f^2\rangle}=k_BT\frac{\left\langle f-\overline{f'}\right\rangle}{\left\langle\left(f'-\overline{f'}\right)^2\right\rangle} \qquad (15)$$

For N CVs $\{f_i\}$ centered at the target value, we get N equations:

$$0{=}\int dX f_i(X)e^{-\beta H_0(X)}\prod_{k=1}^{N}\sum_{j=0}^{\infty}\frac{(-\beta\lambda_k)^j f_k^j}{j!} \qquad (16)$$

At first order, we get the multi-dimensional equivalent expression to equation 15:

$$\boldsymbol{\lambda}{=}k_B T\,\mathrm{Cov}\,(f_i,f_i)^{-1}\cdot\langle\boldsymbol{f}\rangle \qquad (17)$$

This solution is equivalent to taking a single step of Newton's method (see, equation 10 and 11).

These results are tested on the hENM model in Figure 2. In Figure 2D we vary the target value for the twist angle in an actin monomer from above to below the average $\boldsymbol{\phi}$. Using Mathematica, we compute the moments $\left\langle\left(\phi-\overline{\phi}\right)^n\right\rangle$ and solve Eq. 14 numerically truncating from first to fifth order in λ. We find for the distribution of $\boldsymbol{\phi}$ generated by our hENM model, the even-ordered solutions can be purely imaginary. However, the odd ordered equations always have a real solution and these are shown on the right as a function of how far the target is from the unbiased mean, scaled by the standard deviation, $Z{=}(\overline{\phi}-\langle\phi\rangle)/\sqrt{\left\langle(\phi-\langle\phi\rangle)^2\right\rangle}$.

As expected, for target values close to the unbiased average, the bias parameter is very small. We observe that the first order approximation breaks down near Z = ±0.5 and the fifth order solution breaks down around Z = ±1. The linearity of the learned bias parameters in this range is likely a consequence of applying this method to a purely harmonic test system (although the cost of twisting $\boldsymbol{\phi}$ is not perfectly quadradic).

Given the simplicity of computing the first order solution (even for N CVs), we suggest that future practitioners of these relative entropy minimization algorithms start with this as an initial guess. Since it arises from a linear response approximation, this amount of bias is unlikely to cause an irreversible change in the system when activated. However, in the rest of the data below, this is not done so as to show the full process of the learning algorithm.

Simultaneous update of bias parameters outperforms stochastic gradient descent

To fulfill our goal outlined in the introduction, we need to bias the actin monomer to behave as if it is in a filament. To do this, we will attempt to simultaneously bias four CVs: the twist angle and cleft distance, as well as their second moments (fluctuations). As discussed before, the optimal $\boldsymbol{\tau}^{\mathrm{avg}}$ for a CV is related to that CV's auto-correlation time. The problem that can be encountered when going to multiple CVs is that there can be a large timescale separation. This forces a user of the method to use a $\boldsymbol{\tau}^{\mathrm{avg}}$ as big as needed for the slowest-to-average CV. In the stochastic algorithm, only one CV is updated at a time, and so it is possible to have many long averaging periods in a row where only the faster averaging CV's bias parameter is updated. In Figure 3A(top), we first show in simulations of the hENM model that the SGD algorithm is effective for biasing the average cleft distance and twist angle. However, in order to bias these parameters and their second moments, a longer averaging time is needed, making convergence much slower. We show in Figure 3A(bottom) that in this case, using the full covariance matrix as in Eq. 10 greatly accelerates convergence (note that this is after choosing a 10x lower learning rate parameter A for the covariance calculation).

Levenberg-Marquardt algorithm greatly improves on gradient descent

Since we were no longer using the stochastic version of the optimization algorithm, we modified our learning algorithm to include the LM update rule.[28] In Figure 3B, we compare results of LM to the covariance version of our algorithm. We see that the LM algorithm with $\gamma = 0.01$ accelerates convergence by another factor of >10x (~50 vs ~500 ns) over the covariance gradient descent. This allows us to easily see the difference in speed between the two algorithms. This also demonstrates that when using the more “intelligent” step sizes of the LM algorithm, we no longer find it necessary to tune A to get fast convergence. We also show one simulation using the adaptive algorithm described above. In this case, we see that the adaptive LM algorithm converges at about the same speed as the algorithm with its fixed initial γ. However, we note that this method may be useful in atomistic contexts (such as discussed in the following sections), where it could be advantageous to for this this confidence parameter γ to change as the atomistic system moves from one state to another.

Linear bias on monomer CVs is effective as a restraint, matches structure in filament

Having now developed and implemented improved learning algorithms, we sought to test these methods on the all-atom MD simulation of G-actin and F-actin monomers. The four CVs, cleft distance and angle, as well as their second moments, were targeted to match the values from a single actin monomer in the 13-mer simulation shown in Figure 1B (see Table 2). The LM algorithm of all various flavors were found to converge in reasonable amounts of simulation time, with the amount of time required to match all CVs depending upon the particular parameters. In the upper panels of Figure 4A, we show a simulation using the adaptive LM that matches the target CVs in ~100ns, with a small error in all 4 CVs (see Table 2). We note that given the moderately large expense of simulating this system, we did not try to optimize the parameters used much beyond what was learned from the simpler hENM, besides increasing τ^{avg} to account for the slower fluctuations in the atomistic system. Nevertheless, the algorithms showed themselves to be robust, converging eventually in all trials of the LM algorithm with large enough τ^{avg}. The LM results in Figure 4A are compared to a harmonically biased simulation with a very large spring constant applied to both the twist angle and cleft distance CVs. Despite being ~10x larger than what was used previously in a QM/MM application,[3–4] the harmonic bias (starting from the G-actin structure) fails to match the target twist angle. In Figure 4B, we compare the structure of this biased G-actin monomer to a filament subunit configuration. We see that as the algorithm progresses, the G-actin structure (which is already similar to an F-actin monomer structure) converges to a structure that has a lower RMSD to the target, although RMSD was not an observable explicitly biased. We note that previous simulation studies have shown that actin filaments are heterogeneous,[20] and the RMSD of an actin subunit in a filament compared to another or compared to itself later in an MD simulation is typically in this range (for the last

100ns of the simulation in Figure 1B, the subunit-subunit RMSD is 3.35 ± 0.30 Å). Hence, we cannot expect to achieve a closer match than what is seen in Figure 4B without explicitly biasing RMSD.

In the lower panels of Figure 4A, the adaptive LM algorithm is used on a simulation of an actin monomer starting from a flat, filamentous like structure. In this case, it can be seen that around 50 ns, the structure begins to twist and open, as was seen in the unbiased simulation in Figure 1B. Remarkably, the bias algorithm is able to adapt to this change in observable and return the configuration to close to the target value. In other trials using the stochastic gradient descent and covariance gradient descent, the convergence was not fast enough to prevent the structure from twisting, and then it took a much longer time for the system to return to a flattened structure. We note that during this 200ns simulation, convergence is not yet achieved, meaning that there is greater error here/slower convergence to the target CV values than starting from the twisted structure. This seems to be a general trend, and we will consider this idea further in the Discussion section. Finally, in these lower panels we show the result of freezing the bias that has been learned at 80ns from the flat-monomer (before convergence) and show that the simulation approximately maintains the CV values from that starting time.

Biasing CVs in a larger subsystem is an effective alternative constrained moiety

When considering the problem of biasing a subsystem to represent aspects of its behavior within a larger supersystem, in general there will be many possible subsystems of different sizes to consider using. In most cases, it would be advantageous to choose the smallest system size possible, as we have done by choosing a single actin subunit in a filament. However, larger systems afford the advantage of representing a more native-like context; for example, including more protein-protein interfaces. To demonstrate this idea, we tested our algorithms for an actin trimer which was previously used in a metadynamics study[19] of the nucleotide effect on the conformational states of actin (Figure 5A). In Figure 5B, we show data from a simulation where the cleft distance and dihedral angle in all three actin subunits are biased, but in this case only the mean values for these six CVs are set. The algorithm is able to find six parameters for which all the CVs closely match their target values (see Table 2). Interestingly, due to the allosteric coupling between monomers, variances of these six CVs are much closer to the full filament values than in the case of an unbiased monomer. This is reflected in Figure 5C where the dihedral angles of an unbiased trimer simulation are histogrammed and compared to the filament value, with the unbiased dihedral distributions having only ~35% larger standard deviations than an actin subunit in a filament (and the distance distribution being even closer, ~20% different, not shown). When the bias is applied, then, the peak of the distribution shifts as desired while leaving the rest of the distribution approximately the same. The result is only ~10% error in the mean of twist angle and <15% error in the second moment (with distances having <3% error in both). Although a larger system, we believe this kind of tradeoff between size and additional molecular context could be appropriate in some future applications of our GCDS methodology.

Discussion and Conclusions

In this work, we present the idea that linear biases learned by a relative entropy minimization scheme can be used to restrain a molecular system in such a way that it retains some information about its behavior within a larger macromolecular “supersystem” context. In particular, we show that in the case of actin, we can bias two CG observables and their second moments, with the result being that an actin monomer can adopt and maintain a filament-like conformation with native-like fluctuations in these observables. Although the CGDS algorithm takes some time as compared to simply applying a harmonic bias, these methods can achieve a closer match for the distribution of the target observables. We have also made a number of algorithmic improvements which for this application greatly reduce the amount of sampling needed to learn the biasing parameters, down to a very reasonable amount of simulation time (<100ns), and produce systems that would be appropriate for use in subsequent free energy or reactive (QM/MM or otherwise) MD simulations where a small subsystem is required but including the effects of larger scale fluctuations are expected to affect the results.

In previous studies from our group, the stochastic gradient descent algorithm was sufficient to learn biasing parameters in a very short amount of simulation time.[8, 11] We note that the context we are presenting here is very different, where within the simulation the observables to be biased depend on the positions of a few CG observables. In the previous studies, the systems of interest were isotropic liquids, and the CVs of interest (averages over radial distribution functions) depend on the pairwise distance between each molecule. This produces a self-averaging such that the effect of changing the biasing parameter can be sampled over many environments simultaneously. In the protein context, the relaxation time of the observable is much longer and moreover it is likely to be sampled only over a single copy of the system. We suggest that (1) starting from a linear response approximation to the bias parameters, (2) optimizing sampling time, and (3) using methods such as the Levenberg-Marquardt algorithm that take advantage of covariance and try to make “smart” step sizes based on that information, are all important steps that should be applied in this (CGDS) context. Although we suspect these three steps are likely useful in the prior cases, in practice the learning time for previous applications was not a bottleneck.

Previously, there was some concern in the literature that using the full covariance matrix with correlated observables might result in an optimization problem that was non-convex and might not converge.[6] Although in this work we have biased both the first and second moment of our CVs, which are correlated, we have not found this to be a problem in practice (as also discussed in Ref. 9). Another concern with using the full covariance matrix and a Newton-like method with correlated observables is that it might be singular and hence produce divergent step sizes in the iterative algorithm. As noted previously, the Levenberg-Marquardt term proportional to γ (Eq. 10) is specifically designed to avoid this problem.[28]

Finally, we return to one challenge encountered during our CGDS studies, which is that these adaptive algorithms are not well tailored to biasing a system that starts close to a target configuration in a state that is metastable on simulation time scales. An idealized free energy surface in such a case is likely to look something like the illustration in Figure 6. We in fact know this for an actin monomer from previous and ongoing umbrella sampling and metadynamics simulations using these CVs to obtain the free energy landscape for actin flattening.[21, 31] If the starting state of the system is a local free energy minimum, such as for an ATP-bound actin monomer in a flattened configuration, then the initial estimate for the Lagrange multiplier on that CV will be close to zero until fluctuations begin to cause the system to drift towards a deeper minimum. This is exactly what is observed in Figure 4A. On the other-hand, starting in another state is less desirable for two reasons: (1) starting from the lower free energy minimum, the Lagrange multiplier estimate may be very large at first, and then it may take some time once the system is near the target state for the Lagrange multiplier to return to its fixed point, and (2) during this process, there is no guarantee that the simulation will find the target structure as the values of the CVs are improved (although we demonstrate in Figure 4B that this was not an issue for the case of G-actin). This is a challenge for which we do not yet have a complete solution. Yet, we have found in our experience that biasing the second moment of the target CV in addition to the mean when starting from the higher free-energy state goes a long way towards solving this problem. We believe that this is because, as illustrated in Figure 6, even if the minimum for the subsystem has the same mean as the target system, it is likely to be much less constrained, and hence will have a much wider basin in any CG CV such as the ones considered in this work.

The present paper does not claim to provide the final word in the development of CG “guided” or “directed” methods to bias an all-atom MD simulation of a subsystem within a larger supramolecular complex, so that it “feels” certain key effects of being in the macromolecular supersystem, such as a structural bias and altered fluctuations. However, we suggest that the CGDS method developed herein is an important step in that direction. We will certainly improve and extend this approach in the future, and we encourage other researchers to contribute to this effort.

Acknowledgments

The authors thank Professors Giovanni Bussi, Andrew White, and Jonathan Weare for many insightful discussions. GMH was supported by an NIH Ruth L. Kirschstein NRSA Fellowship (F32 GM113415). TDL and GAV were supported by the Office of Naval Research (ONR Grant No. N00014-15-1-2493). Simulations were performed in part using resources provided by the University of Chicago Research Computing Center (RCC), the Department of Defense High Performance Computing Modernization Program, and resources at the San Diego Supercomputer Center provided via the Extreme Science and Engineering Discovery Environment (XSEDE), supported by NSF grant no. ACI-1053575.

References

1. Frenkel, D., Smit, B. Understanding molecular simulation: from algorithms to applications. 2nd. Academic Press; San Diego: 2002. p. xxiip. 638
2. Smith JC, Roux B. Eppur si muove! The 2013 Nobel Prize in Chemistry. Structure. 2013; 21(12): 2102–5. [PubMed: 24315458]
3. McCullagh M, Saunders MG, Voth GA. Unraveling the mystery of ATP hydrolysis in actin filaments. J Am Chem Soc. 2014; 136(37):13053–8. [PubMed: 25181471]
4. Sun R, Sode O, Dama JF, Voth GA. Simulating Protein Mediated Hydrolysis of ATP Other Nucleoside Triphosphates by Combining QM/MM Molecular Dynamics with Advances in Metadynamics. J Chem Theory Comput. 2017; 13(5):2332–2341. [PubMed: 28345907]
5. Bahar I, Rader AJ. Coarse-grained normal mode analysis in structural biology. Curr Opin Struct Biol. 2005; 15(5):586–92. [PubMed: 16143512]

6. Pitera JW, Chodera JD. On the Use of Experimental Observations to Bias Simulated Ensembles. J Chem Theory Comput. 2012; 8(10):3445–51. [PubMed: 26592995]
7. Roux B, Weare J. On the statistical equivalence of restrained-ensemble simulations with the maximum entropy method. J Chem Phys. 2013; 138(8):084107. [PubMed: 23464140]
8. White AD, Voth GA. Efficient Minimal Method to Bias Molecular Simulations with Experimental Data. J Chem Theory Comput. 2014; 10(8):3023–30. [PubMed: 26588273]
9. Cesari A, Gil-Ley A, Bussi G. Combining Simulations Solution Experiments as a Paradigm for RNA Force Field Refinement. J Chem Theory Comput. 2016; 12(12):6192–6200. [PubMed: 27951677]
10. Dannenhoffer-Lafage T, White AD, Voth GA. A Direct Method for Incorporating Experimental Data into Multiscale Coarse-Grained Models. J Chem Theory Comput. 2016; 12(5):2144–53. [PubMed: 27045328]
11. White AD, Knight C, Hocky GM, Voth GA. Communication: Improved ab initio molecular dynamics by minimally biasing with experimental data. J Chem Phys. 2017; 146(4):041102. [PubMed: 28147531]
12. Pollard TD, Cooper JA. Actin a central player in cell shape movement. Science. 2009; 326(5957):1208–12. [PubMed: 19965462]
13. Dominguez R, Holmes KC. Actin structure function. Annu Rev Biophys. 2011; 40:169–86. [PubMed: 21314430]
14. Oda T, Iwasa M, Aihara T, Maeda Y, Narita A. The nature of the globular- to fibrous-actin transition. Nature. 2009; 457(7228):441–5. [PubMed: 19158791]
15. Blanchoin L, Pollard TD. Hydrolysis of ATP by polymerized actin depends on the bound divalent cation but not profilin. Biochemistry. 2002; 41(2):597–602. [PubMed: 11781099]
16. Melki R, Fievez S, Carlier MF. Continuous monitoring of Pi release following nucleotide hydrolysis in actin or tubulin assembly using 2-amino-6-mercapto-7-methylpurine ribonucleoside and purine-nucleoside phosphorylase as an enzyme-linked assay. Biochemistry. 1996; 35(37):12038–45. [PubMed: 8810908]
17. Chu JW, Voth GA. Allostery of actin filaments: molecular dynamics simulations and coarse-grained analysis. Proc Natl Acad Sci U S A. 2005; 102(37):13111–6. [PubMed: 16135566]
18. Pfaendtner J, Lyman E, Pollard TD, Voth GA. Structure dynamics of the actin filament. J Mol Biol. 2010; 396(2):252–63. [PubMed: 19931282]
19. Pfaendtner J, Branduardi D, Parrinello M, Pollard TD, Voth GA. Nucleotide-dependent conformational states of actin. Proc Natl Acad Sci U S A. 2009; 106(31):12723–8. [PubMed: 19620726]
20. Fan J, Saunders MG, Voth GA. Coarse-graining provides insights on the essential nature of heterogeneity in actin filaments. Biophys J. 2012; 103(6):1334–42. [PubMed: 22995506]
21. Saunders MG, Tempkin J, Weare J, Dinner AR, Roux B, Voth GA. Nucleotide regulation of the structure and dynamics of G-actin. Biophys J. 2014; 106(8):1710–20. [PubMed: 24739170]
22. Saunders MG, Voth GA. Comparison between actin filament models: coarse-graining reveals essential differences. Structure. 2012; 20(4):641–53. [PubMed: 22483111]
23. Tribello GA, Bonomi M, Branduardi D, Camilloni C, Bussi G. PLUMED 2: New feathers for an old bird. Comput Phys Commun. 2014; 185(2):604–613.
24. Hocky GM, Baker JL, Bradley MJ, Sinitskiy AV, De La Cruz EM, Voth GA. Cations Stiffen Actin Filaments by Adhering a Key Structural Element to Adjacent Subunits. J Phys Chem B. 2016; 120(20):4558–67. [PubMed: 27146246]
25. Graceffa P, Dominguez R. Crystal structure of monomeric actin in the ATP state. Structural basis of nucleotide-dependent actin dynamics. J Biol Chem. 2003; 278(36):34172–80. [PubMed: 12813032]
26. Abraham MJ, Murtola T, Schulz R, Páll S, Smith JC, Hess B, Lindahl E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX. 2015; 1–2:19–25.
27. Lyman E, Pfaendtner J, Voth GA. Systematic multiscale parameterization of heterogeneous elastic network models of proteins. Biophys J. 2008; 95(9):4183–92. [PubMed: 18658214]

28. Stinis P. A maximum likelihood algorithm for the estimation and renormalization of exponential densities. J Comput Phys. 2005; 208(2):691–703.

29. Ferrarotti MJ, Bottaro S, Perez-Villa A, Bussi G. Accurate multiple time step in biased molecular simulations. J Chem Theory Comput. 2015; 11(1):139–46. [PubMed: 26574212]

30. Wolfram Research I. Mathematica. Champaign, Illinois: 2017. Version 11.1 ed.; Version 11.1

31. Dama JF, Hocky GM, Sun R, Voth GA. Exploring Valleys without Climbing Every Peak: More Efficient and Forgiving Metabasin Metadynamics via Robust On-the-Fly Bias Domain Restriction. Journal of Chemical Theory and Computation. 2015; 11(12):5638–5650. [PubMed: 26587809]

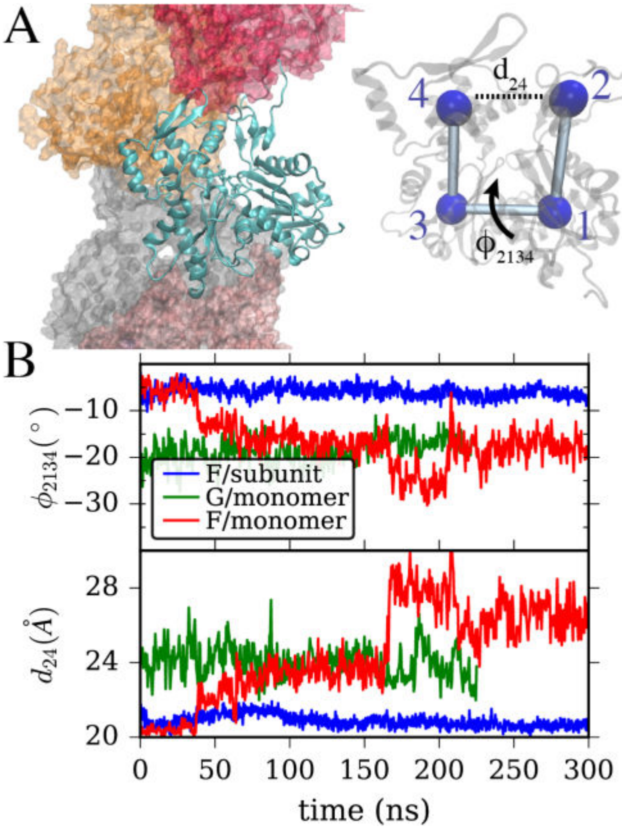


Figure 1.

(A) Left, a snapshot from an actin filament simulation shows one actin subunit in ribbon style with a bound ATP molecule, surrounded by adjacent subunits. Right, a single actin subunit is overlaid with CG beads at the center of mass of its four major subdomains. Important CVs describing the transition from globular to filamentous conformation are the "cleft distance" from bead 2 to 4, and the "twist" dihedral angle formed by the four subdomains, a rotation around the central "bond" as shown. (B) The values of twist angle and cleft distance are shown for three systems. In blue is a single actin subunit within a filament, in green, a single G-actin in solution starting from its crystal structure, and in red, a single actin in solution starting from the filamentous structure.

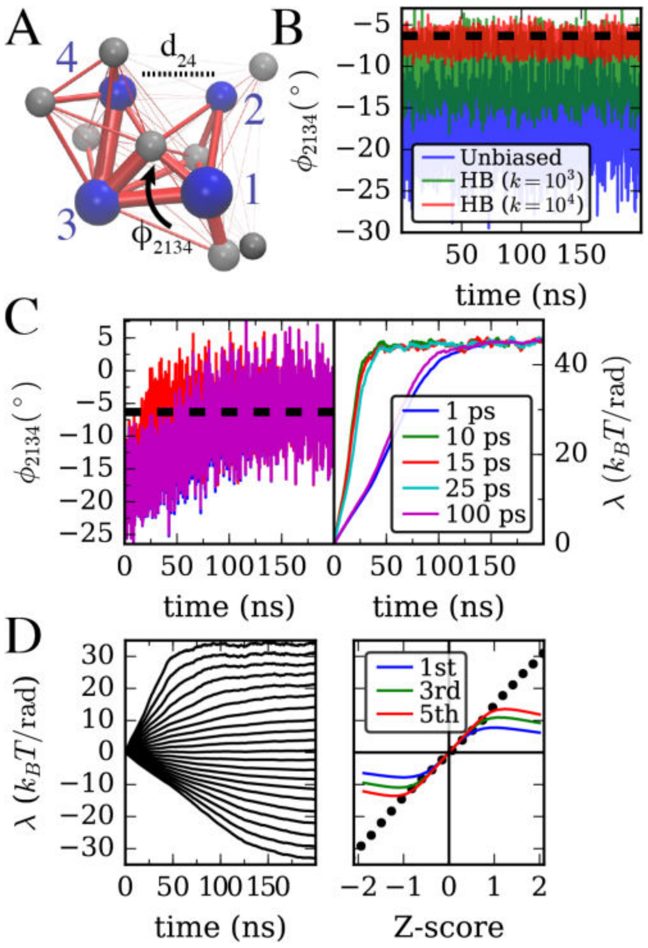


Figure 2.

(A) Twelve-site hENM of an ATP-bound actin monomer parameterized as described in the main text. The four major subdomains of actin are labeled, and cleft distance and twist angle CVs are defined as in Figure 1A. (B) Twist angle for an unbiased hENM, as well as with a harmonic bias with force constants 10^3 and 10^4 kJ/mol/rad^2 centered at $\bar{\phi} = -6.3°$ (dashed line). (C) Twist angle evolution as well as biasing parameter using gradient descent algorithm of Ref. 8 is shown for different τ^{avg}. (D) Left, bias parameter as in C (τ^{avg} = 10ps) with target values for ϕ from −25.2° to −9.16°. Right, comparison of final bias parameters on left (dots) with first, third, and fifth order predictions given in the main text. The horizontal axis shows difference of target ϕ from $\phi_{unbiased}$ scaled by the unbiased standard deviation as computed from data in (B).

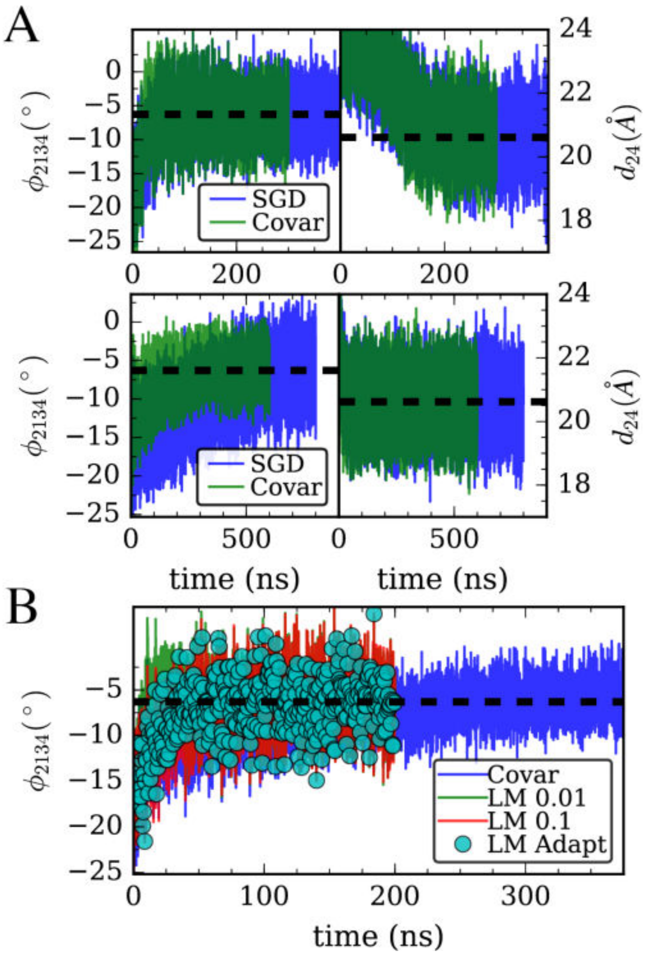


Figure 3.

(A) Simulation of hENM model using stochastic gradient descent (blue) and full covariance matrix (green) to bias the two shown CVs as well as their variance, with otherwise identical algorithmic parameters, (B) The full covariance method is compared to the Levenberg-Marquardt (LM) algorithm with $\gamma = 0.1$, $\gamma = 0.01$, and the adaptive algorithm with starting $\gamma = 0.1$.

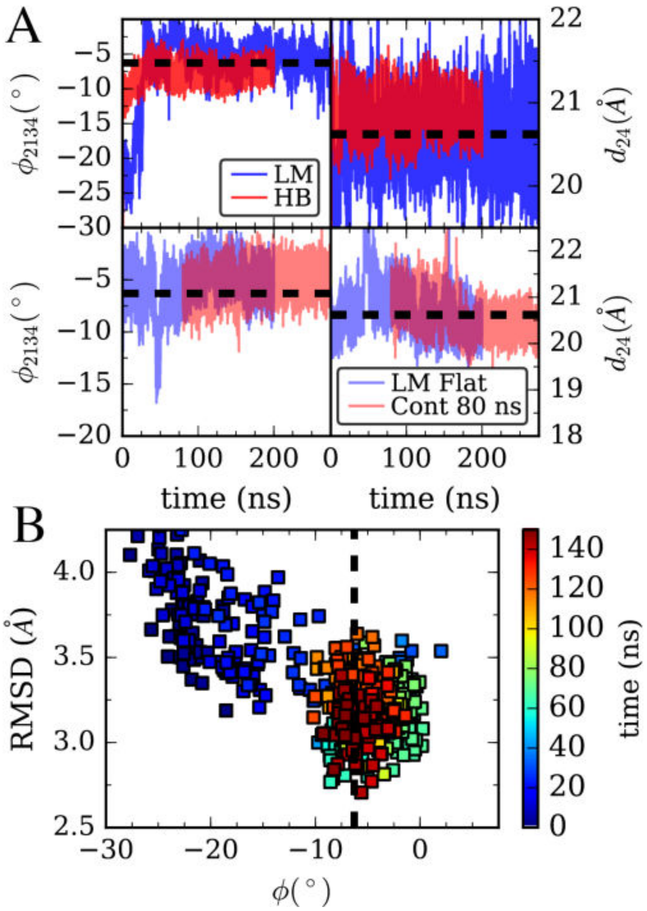


Figure 4.

(A) Top, Adaptive Levenberg-Marquardt (LM) algorithm matching 4 CVs: cleft distance, twist angle, and their second moments is compared to harmonic bias on angle and distance with large spring constants on both. Data is for all-atom MD simulations of the G/monomer system in Figure 1. Bottom, in blue, the LM algorithm is performed on an actin monomer starting from a filament structure (F/monomer in Figure 1). In red, the bias parameters at time 80 ns are fixed and a separate simulation is run using this learned bias. **(B)** Comparison of the structure of the G/monomer in the LM trajectory from (A,top) with a filament subunit by backbone RMSD. Color shows progress along the trajectory in (A).

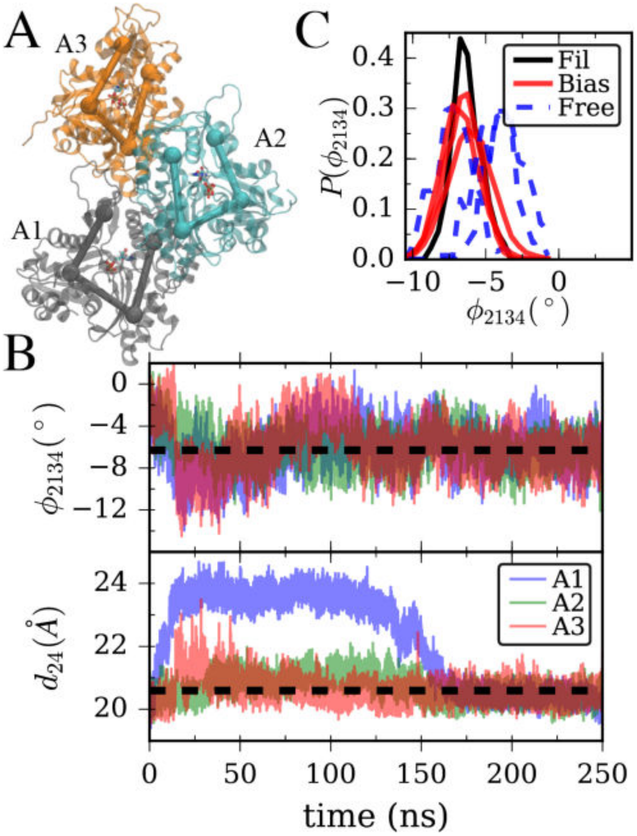


Figure 5.

(A) Illustration of all-atom three actin sub-filament with CG subdomains from Figure 1A overlaid. (B) Twist angle and cleft distance for each of the subunits in (A) during LM bias simulation. (C) Observed distribution of twist angles in the final 50ns of an unbiased 100ns simulation of the structure in (A) (dashed line) vs. the final 50ns of the biased simulation with data plotted in (B) and the final 50 ns of filament data from Figure 1B.

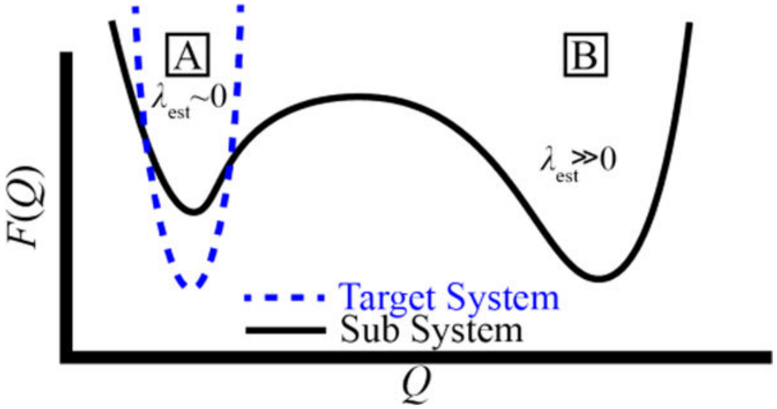


Figure 6.

A structure of a target system (in this study, an actin filament) is likely to be known from experiment, and as such is in a relatively deep local free-energy minimum. Hence, the observed values for a CV (Q) are likely to be normally distributed around a single value (with a roughly harmonic potential of mean force $F(Q)$). When a sub-structure such as an actin monomer is removed to solution, the starting structure (A) will likely still be near a local free energy minimum, however there may be alternative lower free-energy configurations (B). The initially-estimated Lagrange multipliers needed to have the subsystem stay in state A will depend on whether the system starts in state A or B.

Table 1

Algorithmic parameters

Fig/System	Algorithm	Parameters
Fig. 2B/hENM	HB	$k_{\phi}=10^{3}, 10^{4}\frac{\mathrm{kJ}}{\mathrm{mol\,rad}^{2}}, \overline{\phi}=-6.32^{\circ}$
Fig. 2C/hENM	SGD	$A=10k_{B}T, \overline{\phi}=-6.32$ τ^{avg} = from 1–100 ps
Fig. 2D/hENM	SGD	$A=10k_{B}T, \tau^{\mathrm{avg}}=10\mathrm{ps}. \overline{\phi}: -9.16^{\circ}$ to $-25.2^{\circ}(\langle\phi\rangle \approx -17^{\circ}, \sigma_{\phi} \approx 4^{\circ}$
Fig. 3A (top)/hENM	SGD	τ^{avg} = 10 ps, A = 10k$_{B}$T, $\overline{\phi}=-6.32^{\circ}, \overline{d}=20.62\text{Å}$
Fig. 3A (top)/hENM	Covar	See above, but A = 1k$_{B}$T
Fig. 3A (bottom)/hENM	SGD	τ^{avg} = 100 ps, A = 10k$_{B}$T, $\overline{\phi}=-6.32^{\circ}, \overline{\phi^{2}}=40.86^{(\circ)^{2}}, \overline{d}=20.62\text{Å}$, $\overline{d^{2}}=425.03\text{Å}^{2}$
Fig. 3A (top)/hENM	Covar	See above, but A = 1k$_{B}$T
Fig. 3B/hENM	Covar	τ^{avg} = 100 ps, A = 1k$_{B}$T, $\overline{\phi}=-6.32^{\circ}, \overline{\phi^{2}}=40.86^{(\circ)^{2}}, \overline{d}=20.62\text{Å}$, $\overline{d^{2}}=425.03\text{Å}^{2}$
	LM	See above, with γ = 0.1,0.01
	Adaptive LM	See above, with γ^{0} = 0.1, *m* = 3, *l* = 1.2, *LM_stride* = 10
Fig. 4A (top), 4B/G-monomer	HB	$k_{d}=k_{\phi}=5000\frac{\mathrm{kcal}}{\mathrm{mol\,red}^{2}}, \overline{\phi}=-6.32^{\circ}, \overline{d}=20.62\text{Å}$
	Adaptive LM	τ^{avg} = 100 ps, A = 10k$_{B}$T, γ^{0} = 0.1 $\overline{\phi}=-6.32^{\circ}, \overline{\phi^{2}}=40.86^{(\circ)^{2}}, \overline{d}=20.62\text{Å}, \overline{d^{2}}=425.03\text{Å}^{2}$, *m* = 3, *l* = 1.2, *LM_stride* = 100
Fig. 4A (bottom)/F-monomer	Adaptive LM	See above, except γ^{0} = 0.1
	Frozen bias (after 80 ns of above)	$\lambda_{\phi}=-10.2\frac{\mathrm{kcal}}{\mathrm{mol\,rad},} \lambda_{\phi^{2}}=484.0\frac{\mathrm{kcal}}{\mathrm{mol\,rad}^{2},} \lambda_{d}=2.2\frac{\mathrm{kcal}}{\mathrm{mol\,Å},}$ $\lambda_{d^{2}}=6.63\frac{\mathrm{kcal}}{\mathrm{mol\,Å}^{2}}$
Fig. 5B,C	LM	τ^{avg} = 100ps, A = 1k$_{B}$T, γ^{0} = 0.01 $\overline{\phi_{A1}}=\overline{\phi_{A2}}=\overline{\phi_{A3}}=-6.32^{\circ}, \overline{d_{A1}}=\overline{d_{A2}}=\overline{d_{A3}}=20.6\text{Å}$

Table 2

Observed values for collective variables parameters

Quantities are computed for the final 50ns shown in each figure. Percentages are comparison with respect to a single actin monomer, data on the first line of the table (bold). Biased parameters are underlined.

Fig/System+method	$\langle d_{24}\rangle$(Å)	$\langle d_{24}^2\rangle$(Å^2)	$\langle \phi_{2134}\rangle(\circ)$	$\langle \phi_{2134}^2\rangle(\circ^2)$
Fig. 1B/F-actin (1 subunit)	**20.62 (0.0%)**	**425.03 (0.0%)**	**−6.32 (−0.0%)**	**40.86 (0.0%)**
Fig. 1B/G-monomer	23.82 (15.5%)	568.15 (33.7%)	−16.93 (168.0%)	290.70 (611.4%)
Fig. 1B/F-monomer	26.52 (28.6%)	704.00 (65.6%)	−17.37 (175.0%)	307.00 (651.3%)
Fig. 2B/hENM (no bias)	23.95 (16.2%)	574.24 (35.1%)	−17.44 (176.1%)	317.18 (676.2%)
Fig. 2B/hENM HB K=1000	23.79 (15.4%)	566.72 (33.3%)	−10.54 (66.8%)	116.27 (184.5%)
Fig. 2B/hENM HB K=10000	23.82 (15.5%)	568.04 (33.6%)	−6.95 (9.9%)	49.07 (20.1%)
Fig. 2C/hENM SGD τ^{avg} = 15 ps	23.88 (15.8%)	570.98 (34.3%)	−6.30 (−0.2%)	54.97 (34.5%)
Fig. 3A/hENM SGD 2 CV	20.62 (−0.0%)	425.72 (0.2%)	−6.48 (2.6%)	52.02 (27.3%)
Fig. 3A/hENM Covar 2 CV	20.65 (0.2%)	427.28 (0.5%)	−6.53 (3.4%)	52.65 (28.8%)
Fig. 3A/hENM SGD 4 CV	20.60 (−0.1%)	425.03 (−0.0%)	−6.50 (2.8%)	50.99 (24.8%)
Fig. 3A&B/hENM Covar 4 CV	20.60 (−0.1%)	424.88 (−0.0%)	−5.99 (−5.2%)	40.03 (−2.0%)
Fig. 3B/hENM LM γ = 0.1	20.62 (0.0%)	425.95 (0.2%)	−6.04 (−4.3%)	44.52 (8.9%)
Fig. 3B/hENM LM γ = 0.01	20.63 (0.0%)	426.02 (0.2%)	−6.07 (−4.0%)	44.63 (9.2%)
Fig. 3B/hENM LM Adapt γ^0 = 0.1	20.62 (0.0%)	425.94 (0.2%)	−6.02 (−4.7%)	43.97 (7.6%)
Fig. 4A/G-monomer HB 2 CV	20.87 (1.3%)	435.78 (2.5%)	−7.37 (16.7%)	55.38 (35.5%)
Fig. 4A/G-monomer LM Adapt 4 CV	20.55 (−0.3%)	422.51 (−0.6%)	−6.04 (−4.4%)	38.03 (−6.9%)
Fig. 4A/F-monomer LM Adapt 4 CV	20.44 (−0.8%)	418.08 (−1.6%)	−5.11 (−19.1%)	28.01 (−31.5%)
Fig. 4A/F-monomer Fixed Bias 4CV	20.32 (−1.4%)	412.95 (−2.8%)	−4.37 (−30.9%)	20.56 (−49.7%)
Fig. 5/Trimer-A1 LM Adapt	20.32 (−1.4%)	412.90 (−2.9%)	−6.21 (−1.7%)	40.78 (−0.2%)
Fig. 5/Trimer-A2 LM Adapt	20.53 (−0.4%)	421.68 (−0.8%)	−7.03 (11.3%)	51.62 (26.3%)
Fig. 5/Trimer-A3 LM Adapt	20.53 (−0.4%)	421.39 (−0.9%)	−6.75 (6.8%)	47.31 (15.8%)